NPS-EC-97-007

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Generating and Demodulating M-ary CPFSK Using the FFT

by

Paul H. Moose

March 5, 1997

Approved for Public Release; Distribution Unlimited.

Prepared for: NCCOSC RDTE Division

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Rear Admiral M.J. Evans
Superintendent

R. Elster
Provost

This report was sponsored by NCCOSC RDTE Division.

Approved for Public Release; Distribution Unlimited.

The report was prepared by:

PAUL H. MOOSE
Associate Professor
Department of Electrical and
Computer Engineering

Reviewed by:

HERSCHEL H. LOOMIS, JR.
Chairman
Department of Electrical and
Computer Engineering

Released by:

DAVID W. NETZER
Associate Provost and
Dean of Research

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 5, 1997	Research (Technical) 1 Oct 96-1 Jan 97

4. TITLE AND SUBTITLE
Generating and Demodulating M-ary CPFSK Using the FFT

5. FUNDING NUMBERS
N0001497WX20022AA

6. AUTHOR(S)
Paul H. Moose

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER
NPS-EC-97-007

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
NCCOSC RDTE Division
53560 Hull Street
San Diego, CA 93152-52001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this report are those of the author and do not reflect the official policy or position of the Department of Defense or the United States Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited.

12b. DISTRIBUTION CODE
A

13. ABSTRACT (Maximum 200 words)

This paper discusses a technique for modulating and demodulating M-ary CPFSK using an FFT based modem typical of Coded Orthogonal Frequency Division Modulation (COFDM), also known as Discrete Multi-Tone (DMT), systems. COFDM is one of the more promising spectrally efficient, high data rate modulation techniques for mobile digital communications. This paper shows that legacy CPFSK radios like the AN/GRC-226 (binary CPFSK at 256, 512, 1024, and 2048 kbps) used by the U.S. Army could be easily implemented in a DMT modem orginally designed for higher data rate and better spectral efficiency than the legacy radio. MATLAB code is included that simulates the modem.

14. SUBJECT TERMS
digital communications, modem, FFT

15. NUMBER OF PAGES
32

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR

NSN 7540-01-280-5500

STANDARD FORM 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18 298-102

Generating and demodulating M-ary CPFSK using the FFT.

Paul H. Moose

Naval Postgraduate School

Monterey, CA.

March 5, 1997

1.Introduction

The inverse FFT can be used to generate M-ary, full response, continuous phase, frequency shift keying (CPFSK) and the FFT can be used to demodulate the received signal in the following way. We begin by recognizing that an inverse DFT creates N digital carrier frequencies with amplitudes and phases determined by the frequency domain vector X of length N. M-ary CPFSK requires that for each symbol, one of $M=2^q$ carrier frequencies (with initial phase equal the terminal phase of the previous symbol) be transmitted for each symbol where q bits are sent per symbol. Therefore, for each symbol, X is filled with all zeros except a complex modulation value with amplitude one and phase determined by the terminal phase of the previous symbol is placed in the position corresponding to the frequency to be sent for the input symbol. M of the N possible frequencies will be used, but only one frequency will be used on any given symbol.

The receiver simply computes the FFT of the received symbols and extracts from this vector of length N a vector of length M of the positions corresponding to the M transmit frequencies. The FFT amounts to implementation of a quadrature correlator for each of the N frequencies. We select from those the M that could have possible been sent and proceed to decode M-ary CPFSK in accordance with established principles. We shall refer to this type of modem as a discrete multi-tone (DMT) modem. The same modem may be used for OFDM and M-ary FSK [Ref 1]. Diagrams of the transmitter and receiver are shown in Fig. 1.1 and Fig. 1. 2 below.

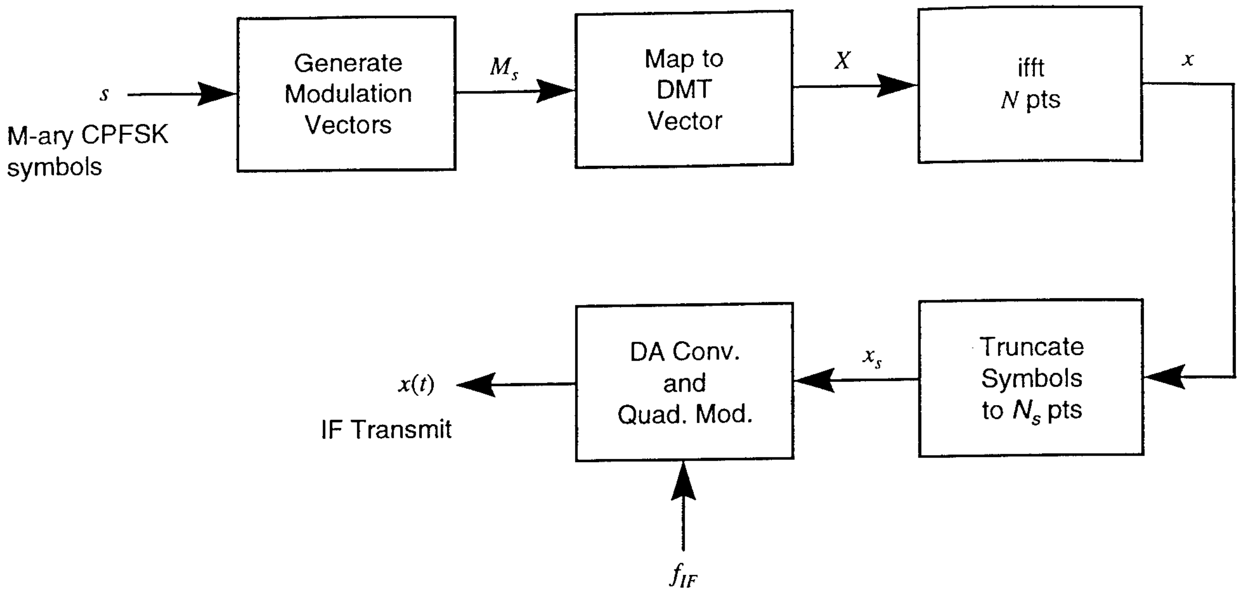

Fig. 1.1 DMT Transmit

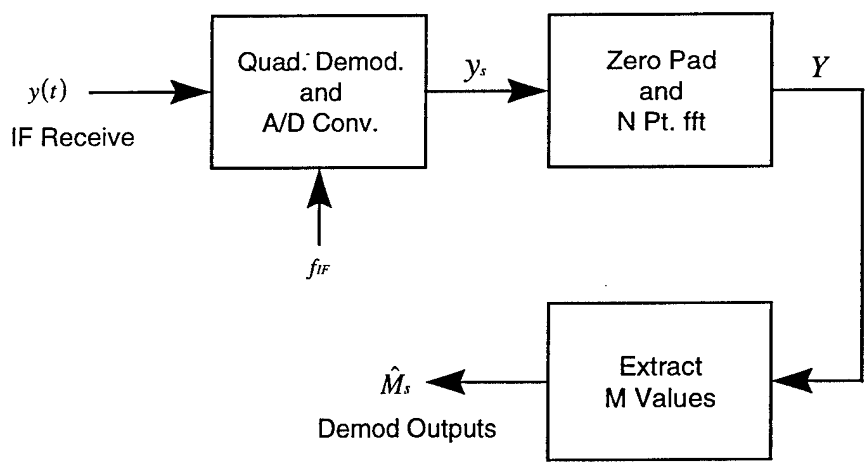

Fig. 1.2 DMT Receive

2. M-ary CPFSK

Let {s} be data symbols from the M-ary alphabet 0,1,2,.......,M-1 and let the information symbols corresponding to these be given by

$$I = 2*s - (M-1) \tag{2.1}$$

such that the {I} are from the alphabet ±1, ±3, ±,5,.....±(M-1). In order to represent a CPFSK symbol, we begin with the PAM signal

$$d(t) = \sum I_n u(t-nT_b). \tag{2.2}$$

Here u(t) is a rectangular pulse of amplitude $1/(2T_b)$ and duration T_b, the symbol interval. [1]

Let

$$\phi(t;I) = 2\pi\ T_b \text{del\_f} [\ \int_0^t d(\tau)\ d\tau\ ] \tag{2.3}$$

be the time-varying phase of the carrier which, since the integral of d(t) is piecewise continuous, is piecewise continuous. The baseband CPFSK transmission waveform is

$$s(t) = A \exp\{j\ \phi(t;I)\ \} \tag{2.4}$$

and consists of a sequence of discrete frequencies, that is during the n^{th} symbol interval,

$$f_n = [d\{\phi(t;I_n)\}/dt]/(2\pi) = [\ \text{del\_f}\ ]I_n/2\ , \qquad n\,T_b \le t < (n+1)T_b. \tag{2.5}$$

---

[1] A rectangular pulse of length T_b generates "full-response" CPFSK. Other pulse shapes generate continuous phase signals referred to as CPM for continuous phase modulation. Longer pulses may be used, rectangular or otherwise and they will generate "partial response" CPFSK or CPM.

The $\{\mathbf{f}\}$ are contained in the alphabet ±del_f/2, ±3del_f/2, ... , ±(M-1)del_f/2. We see that the frequency spacing between any two symbols is a multiple of del_f, the phase of the waveform is piecewise continuous and the amplitude is constant. The phase of the waveform during the n^{th} interval, which is linear for CPFSK, can be determined by evaluation of (2.3) to yield

$$\phi(t;I) = \theta_n + 2\pi h\, I_n\, q(t-nT_b) \tag{2.6}$$

where

$$\theta_n = \pi h \sum_{m=0}^{n-1} I_m \quad , \tag{2.7}$$

is the terminal phase at the end of the $(n-1)^{th}$ interval,

$$h = del\_f^{*}\ T_b = del\_f\,/\,r_b \tag{2.8}$$

is the modulation index and,

$$q(t) = \begin{cases} 0 & t<0 \\ t/(2T_b) & 0 \le t < T_b \\ ½ & T_b \le t \end{cases} \tag{2.9}$$

is the integral of the PAM basic pulse u(t). Thus during the n^{th} interval we must transmit the frequency $f_n = [d\{\phi(t;I_n)\}/dt]/(2\pi) = h\, I_n/(2T_b) =$ [del_f]I_n /2 with initial phase θ_n .

The baseband waveform during the n^{th} interval from (2.4) is

$$s(t) = \exp\{j\, \phi(t;I)\} = \exp\{j\, \theta_n\} \exp\{2\pi j\, f_n\, t\} \tag{2.10}$$

To demodulate CPFSK we compute during each interval the complex correlations for each of the possible transmit frequencies. In the presence of AWGN we receive during the n^{th} interval the signal with complex modulation envelope

$$r(t) = (2\, E_b/\, T_b)^{1/2} \exp\{j\, \phi(t;I)\} + n(t) \tag{2.11}$$

where E_b is the symbol energy and n(t) is the complex baseband AWN with PSD equal N_o for the statistically independent real and imaginary parts. We now compute the complex correlations for each of the M possible transmit frequencies[2]. The output of the k^{th} correlator for the n^{th} interval is

$$z_k = (1/T_b) \int_0^{T_b} r(t) \exp\{-\, 2\pi j\, f^{(k)}\, t\, \}\, dt$$
$$= (2\, E_b/\, T_b)^{1/2} \exp\{j\, \theta_n\, \} \exp\{\pi j\, h\, [I_n - I^{(k)}]\, /2\} \operatorname{sinc}\{\pi h\, [I_n - I^{(k)}]/2\, \}$$
$$+\, n_k \quad , \qquad k=0,1,2,\ldots\ldots ,M\text{-}1, \tag{2.12}$$

where the n_k are complex gaussian random variables with statistically independent real and imaginary parts that have zero means and equal variances N_o/T_b. For $I_n = I^{(k)}$, the correlator output is just

$$z_k = (2\, E_b/\, T_b)^{1/2} \exp\{j\, \theta_n\, \} + n_k \tag{2.13}$$

with mean output $(2\, E_b/\, T_b)^{1/2} \exp\{j\, \theta_n\}$ .The mean outputs of the other correlators depend on h and $I_n - I^{(k)}$. However, note that $I_n - I^{(k)}$ is always a multiple of two, so that for integer values of h, that is when the frequency spacing del_f is a multiple of the symbol rate r_b, all the other correlator mean outputs are equal to zero, that is we have

[2] The real and imaginary parts of the complex correlator output are equal to the in phase and quadrature outputs of a quadrature correlator at frequency $f_o + f^{(k)}$.

orthogonal signaling. We shall discuss the special case of $h = ½$, which is called minimum shift keying (MSK) shortly.

2.1 Phase states [Ref 2]

Consider the sequence of phases generated by (2.7), the initial phases for each symbol. For any given input sequence of data, they form a phase trajectory.
If h is a rational fraction, $h = m/p$, there will be a finite number of phases (modulo 2π) that can occur. These are the phase states of the system.(If h is irrational, there can be an infinite number of states.) If m is an even number, there will be p phase states

$$\{\Theta_s\} = \{\, 0, \pi m/p, 2\pi m/p, 3\pi m/p, \ldots, (p-1)\pi m/p \,\} \tag{2.14a}$$

and if m is odd there will be 2p phase states

$$\{\Theta_s\} = \{\, 0, \pi m/p, 2\pi m/p, 3\pi m/p, \ldots, (2p-1)\pi m/p \,\}. \tag{2.14b}$$

2.1.1 MSK

Consider the case of binary CPFSK with $h = ½$, the modulation index corresponding to MSK. Since $h=\text{del\_f}/r_b$, we note that the bit rate is twice the frequency spacing. Here, since $m = 1$ is odd, there are four phase states

$$\{\Theta_s\} = \{\, 0, \pi/2, \pi, 3\pi/2 \,\}\,, \tag{2.14c}$$

the message alphabet is

$$\{\mathbf{I}\} = \{\, -1, 1\} \tag{2.15}$$

and the frequency alphabet is

$$\{\mathbf{f}\} = \{ \text{-del\_f/2, del\_f/2} \}. \tag{2.16}$$

Assuming the initial phase state is 0, then after the first symbol, the phase, $\theta_2 = \pi h\ I_1$, can be $\pi/2$ or $3\pi/2$. After the second symbol, the phase $\theta_3 = \theta_2 + \pi h\ I_2$ can be 0, or π. The sequence of phase states can be depicted as a trellis as shown in Fig. 2.1.

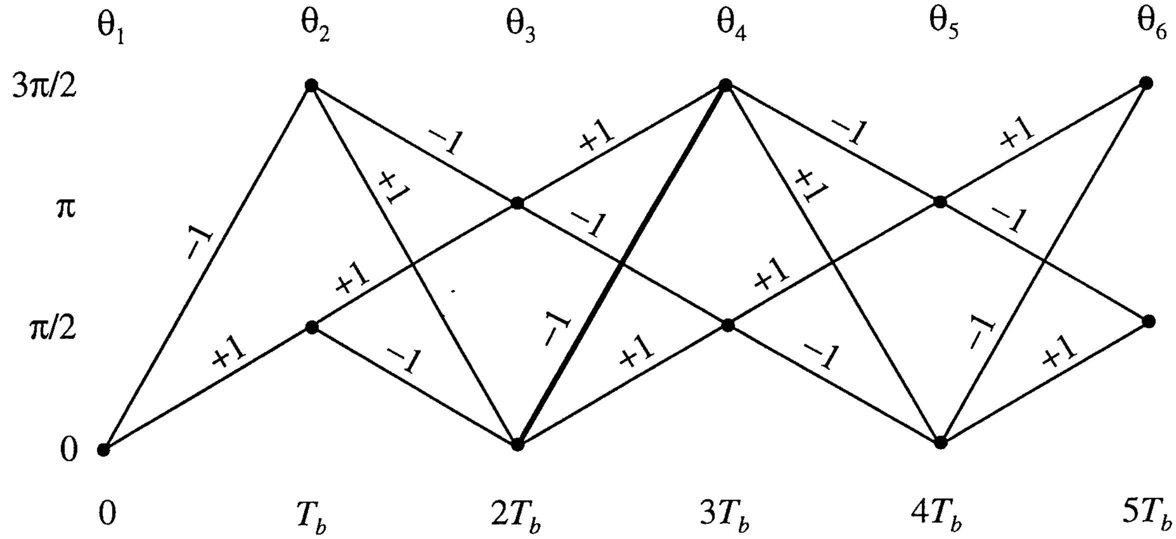

Fig. 2.1 MSK Phase Trellis

2.1.2 Viterbi decoding

The bit stream for CPFSK with a finite number of phase states can be optimally decoded using the Viterbi algorithm [Ref 3]. The Viterbi algorithm finds the most probable path through the trellis diagram in an efficient manner [Ref 4]. Consider Fig. 2.1, the trellis for MSK. At the end of odd numbered symbols, e.g. symbol number three, the system will be in the next even numbered state, in this case state Θ_4, which can be either $\pi/2$ or $3\pi/2$. If it is in state $\theta_4 = \pi/2$, it could have arrived there by a transition from state $\theta_3 = \pi$ with an $I_3 = -1$, or from state $\theta_3 = 0$ with an $I_3 = 1$. If it is in state $\theta_4 = 3\pi/2$, it could have arrived there by a transition from state $\theta_3 = \pi$ with an $I_3 = 1$, or from state $\theta_3 = 0$ with an $I_3 = -1$.

Consider the outputs of the two correlators in the presence of AWGN for MSK, h=1/2. From (2.12)

$$z_0 = (2\, E_b/\, T_b)^{1/2} \exp\{j\, \theta_3\} \exp\{\pi j\, [I_3 - (-1)]\, /4\} \operatorname{sinc}\{ -\pi\, [I_3 - (-1)]/4 \} + n_0 \quad (2.17)$$

and

$$z_1 = (2\, E_b/\, T_b)^{1/2} \exp\{j\, \theta_3\} \exp\{\pi j\, [I_3 - (1)]\, /4\} \operatorname{sinc}\{ -\pi\, [I_3 - (1)]/4 \} + n_{j1} \quad . \qquad (2.18)$$

Note that when $I_3 = (-1)$,

$$E[z_0] = (2\, E_b/\, T_b)^{1/2} \exp\{j\, \theta_3\} \qquad (2.19)$$

and

$$E[z_1] = -j\, (2\, E_b/\, T_b)^{1/2} \exp\{j\, \theta_3\}(2/\pi) \qquad (2.20)$$

and when $I_3 = (1)$,

$$E[z_0] = j\, (2\, E_b/\, T_b)^{1/2} \exp\{j\, \theta_3\}(2/\pi) \qquad (2.21)$$

and

$$E[z_1] = (2\, E_b/\, T_b)^{1/2} \exp\{j\, \theta_3\}. \qquad (2.22)$$

In order to update the path metric for state $\theta_4 = \pi/2$, we add to the path metric at state $\theta_3 = \pi$ the Euclidean distance $d_{0|\pi}$ from z_0 to the expected value of z_0 with $\theta_3 = \pi$ and $I_3 = -1$. From (2.19) the expected value is $-\, (2\, E_b/T_b)^{1/2}$. We add to the path metric for state $\theta_3 = 0$ the Euclidean distance[3] $d_{1|0}$ from z_1 to the expected value of z_1 with $\theta_3 = 0$ and $I_3 = 1$ which from (2.22) is $(2\,E_b/\,T_b)^{1/2}$. We select the minimum of the two total path metrics and retain the path with the minimum metric, deleting the other path and assign the minimum metric to be metric for state $\theta_4 = \pi/2$.

Similarly, in order to update the path metric for state $\theta_4 = 3\pi/2$, we add to the path metric at state $\theta_3 = \pi$ the Euclidean distance $d_{1|\pi}$ from z_1 to the expected value of z_1 with $\theta_3 = \pi$ and $I_3 = 1$, which from (2.22) is $-(2\,E_b/\,T_b)^{1/2}$, and we add to the path metric for state $\theta_3 = 0$ the Euclidean distance $d_{0|0}$ from z_0 to the expected value of z_0 with $\theta_3 = 0$ and $I_3 = -1$ which from (2.19) is $(2\,E_b/\,T_b)^{1/2}$, select the minimum of the two path metrics and retain the path with the minimum metric, deleting the other path and making the minimum metric, the metric for state $\theta_4 = 3\pi/2$.

The four Euclidian distances are illustrated in Fig. 2.2 for the last case, where the system actually transitioned from state $\theta_3 = 0$ to state $\theta_4 = 3\pi/2$ with an $I_3 = -1$. (See Fig. 2.1) In this case, the expected value of z_0, from (2.19), is $(2\,E_b/\,T_b)^{1/2}$ and the expected value of z_1, from (2.20), is $-j\,(2\,E_b/\,T_b)^{1/2}(2/\pi)$.

The update procedure following even numbered symbols intervals follows the same pattern, however, the expected values are rotated by $\pi/2$. Note that at each stage for MSK there are only two phase states to update, even though the system contains four phase states. In fact the four state system can be collapsed to a two state system simply by multiplying the correlator outputs by $\exp\{jn\pi/2\}$. This reduces the path memory requirements for the soft Viterbi decoder by a factor of two but the computational complexity remains the same [Ref 3].

---

[3] Selecting the path with the minimum Euclidean distance as a metric is equivalent to selecting the path with the maximum aposteriori probability in AWGN.

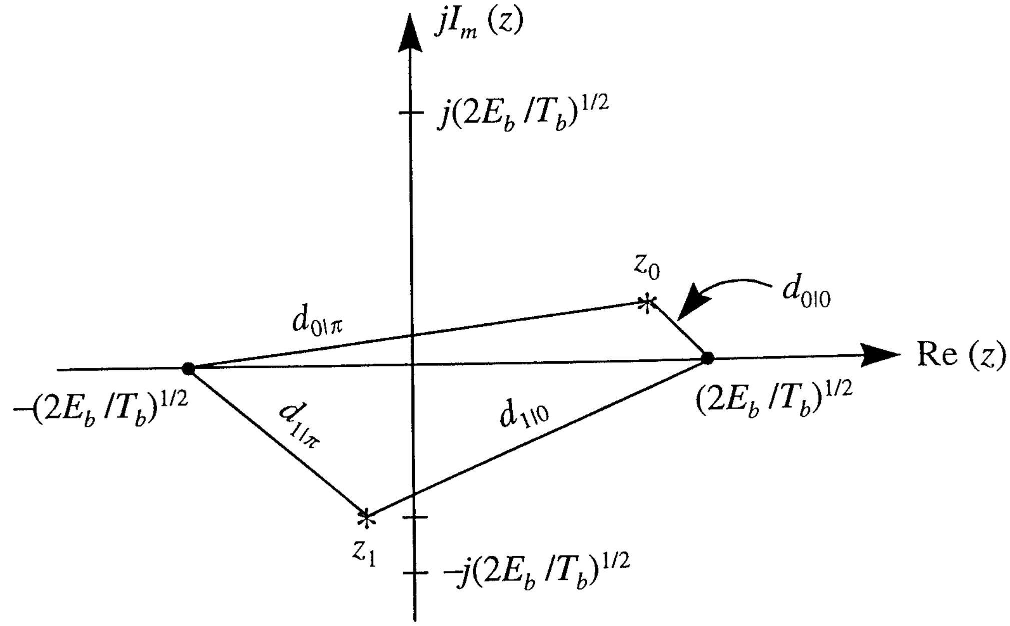

Fig. 2.2 Euclidean distances for soft Viterbi decoding of MSK.

3. Frequency spacing and symbol rate

For CPFSK, the modulation index h = del_f/r_b (see (2.8)) is typically, but not necessarily, less than one. Let T_o = 1/del_f be one period of the inter-carrier spacing del_f. Then

$$T_b = h * T_o \tag{3.1}$$

where T_b is the symbol interval. Now let

$$P = \text{ceil}(h) \tag{3.2}$$

so that $PT_o \geq T_b$ gives an integer number of cycles of del_f with a length greater than or equal to the symbol interval. The frequencies {**f**} needed for the CPFSK are in the alphabet ±del_f/2, ±3del_f/2, ... , ±(M-1)del_f/2. The lowest frequency required has period $2T_o$. Now let

$$T = P^*(\ 2T_o) = N^*\Delta t = 1/\Delta f \tag{3.3}$$

be the period of the lowest frequency, Δf, available from an N point ifft with complex samples clocked out at sampling frequency $f_s = 1/\Delta t$. The frequencies available from the ifft are 0, $\pm\Delta f$, $\pm 2\Delta f$, $\pm 3\Delta f$, ... $\pm[(N/2)-1]\Delta f$. The frequencies required are

$$\{\ \pm del\_f/2, \pm 3del\_f/2, \ldots\ldots , \pm(M-1)del\_f/2\} =$$
$$\{\ \pm P^*\Delta f,\ \pm 3P^*\Delta f, \ldots\ldots, \pm(M-1)P^*\Delta f\}, \tag{3.4}$$

that is P multiples of the lowest frequency. (Note that for h<= 1, P=1). The required symbol length is

$$T_b = h^*T_o = h^*T/(2^*P) \tag{3.5}$$

which requires that we clock out

$$N_s = h^*N/(2^*P) \tag{3.6}$$

points of the N points in the ifft. For example, for h=1/2, P =1, del_f/2 = Δf and N_s =N/4, that is, we clock out 1/4 of the points in the ifft and the phase of the carrier will change by $\pi/2$, which is correct for MSK. A MATLAB program that generates M-ary CPFSK as well as M-ary FSK and OFDM is included as Appendix I.

3.1 Rational modulation index

If the modulation index h = m/p is rational, then

$$N_s = m^*N/(2^*P^*p) \tag{3.7}$$

can always be made an integer by proper choice of N. If h is an integer, then P = h and N_s = N/2. If h is less than one, then P=1 and

$$N_s = m*N/(2*p) \tag{3.8}$$

which will be an integer for N a power of two when p is a power of two for N> 2p. Recall from (2.13) and (2.14) that there are p or 2p phase states depending on whether m is even or odd. Viterbi decoders with states equal a power of two is common practice for convolutional code decoders [Ref 4]. Also for the Army's AN/GRC-226 CPFSK radio, h=1/2 with bit rates of 256,512,1024, and 2048 Kbps, so for this radio, 1/4 of the ifft points are used.

The point of this discussion is that for many practical cases of CPFSK, the required number of discrete time points in the CPFSK symbol specified by (3.6) can easily be made an integer for N a power of two as required by the radix two fft algorithm.

3.2 Frequency spacing error

In the event that (3.6) is not an integer for the specified h, then we can round (3.6) to the nearest integer according to

$$N_s = \text{round}[\, h*N/(2*P)]. \tag{3.9}$$

As illustrated by the timing diagram in Fig. 3.1, if we wish to keep the symbol rate r_b constant then the modulation index is changed from h to h' where

$$h' = 2*P*N_s /N \tag{3.10}$$

and the frequency spacing is changed from del_f to del_f' where

$$del\_f' = h'*r_b \quad . \tag{3.11}$$

The error introduced by this roundoff procedure is analyzed in [Ref 1]. It is inversely proportional to N and can be made negligible for moderate values of N (~ 32).
Notice that (3.10) guarantees a rational h'=m/p with p at most N/2, so there are at most N phase states. For example, 32 phase states is not large for a Viterbi decoder.

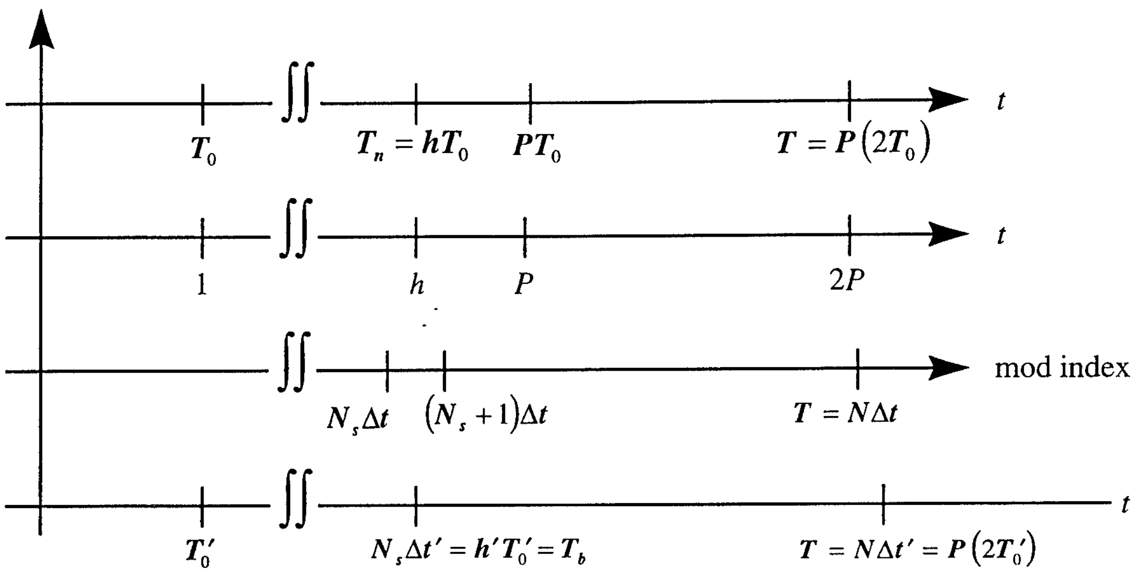

Fig. 3.1 Timing relationships

4. FFT receiver for CPFSK

The receiver for CPFSK samples the baseband received signal at the sample rate f_s, generating N_s samples for each received symbol. These are padded with zeros to length N and the N point fft is computed. The output values found in the frequency cells

$$\{ \pm P*\Delta f, \ \pm 3P*\Delta f, \ ...., \pm(M-1)P*\Delta f\} \tag{4.1}$$

are extracted. These are found in the the fft output vector locations

$$\{(P, N-P), (3P, N-3P), \ ....((M-1)P, N-(M-1)P)\} \tag{4.2}$$

and the values are the discrete time complex correlator outputs. A MATLAB program implementing a fft based receiver for M-ary CPFSK as well as M-ary FSK and OFDM is included as Appendix II.

4.1 Discrete time complex correlation

Recall from section 2 that in the presence of AWGN we receive during the n^{th} symbol interval the signal with complex modulation envelope

$$r(t) = (2\ E_b/\ T_b)^{1/2} \exp\{j\ \theta_n\} \exp\{2\pi j\ f_n\ t\} + n(t) \tag{4.3}$$

where E_b is the symbol energy and n(t) is the complex baseband AWN with PSD equal N_o for the statistically independent real and imaginary parts The discrete time output of the sampler after padding with zeros is

$$r(m) = \begin{cases} (2\ E_b/\ T_b)^{1/2} \exp\{j\ \theta_n\} \exp\{2\pi j\ f_n\ \Delta t\ m\} + n(m) & 0 \le m \le N_s-1 \\ 0 & N_s \le m \le N-1 \end{cases} \tag{4.4}$$

where

$$f_n = I_n\ del\_f/2 = I_n\ P*\Delta f = I_n\ P/(N*\Delta t)$$

Now consider the output of the fft in the frequency cell $f^{(k)} = I^{(k)}P*\Delta f$ (for convenience, we scale by $1/N_s$),

$$z_k = (1/N_s) \sum r(m)\exp(-2\pi j\ m\ I^{(k)}P/N) =$$

$$(2\ E_b/\ T_b)^{1/2} \exp\{j\ \theta_n\}(1/N_s)[\{1-\exp[\ 2\pi j N_s\ (I_n-I^{(k)})\ P/N]\}/\{1-\exp[\ 2\pi j\ (I_n-I^{(k)})\ P/N]\}] + n_k \quad . \qquad (4.5)$$

This should be compared to (2.12), the analog complex correlation result. Note that in the frequency cell corresponding to the transmitted symbol, $I_n=I^{(k)}$, and

$$z_k = (2\ E_b/\ T_b)^{1/2} \exp\{j\ \theta_n\}+ n_k \qquad (4.6)$$

as in (2.13) for the analog correlator. However, in the other non-signaling cells, there may be a slight variation due to the discrete nature of the signal and the zero padding. Using (3.6), we may re-write (4.5) as

$$z_k = (2\ E_b/\ T_b)^{1/2} \exp\{j\ \theta_n\}(1/N_s)[\{1-\exp[\pi jh\ (I_n-I^{(k)})]\}/\{1-\exp[\ \pi jh\ (I_n-I^{(k)})/N_s]\}] + n_k \quad . \qquad (4.5)$$

First consider the case of orthogonal signaling with h an integer. Since $I_n-I^{(k)}$ is a multiple of two for all $I_n \neq I^{(k)}$, then $E[z_k]= 0$ for all $I_n \neq I^{(k)}$ and the digital complex correlator outputs are orthogonal as are those for the analog correlator as in (2.12). However, consider the case of MSK with h=1/2 and binary CPFSK with $I_n-I^{(k)}=\pm 2$ for $I_n \neq I^{(k)}$. Then

$$E[z_1]= (2\ E_b/\ T_b)^{1/2} \exp\{j\ \theta_n\}(1/N_s)[2/\{1-\exp[-\ \pi j/N_s]\}] \quad , \qquad \text{if } I_n = -1 \qquad (4.6)$$

and

$$E[z_o]= (2\ E_b/\ T_b)^{1/2} \exp\{j\ \theta_n\}(1/N_s)[2/\{1-\exp[\ \pi j/N_s]\}] \quad , \qquad \text{if } I_n =1 \qquad (4.7)$$

These should be compared to (2.20) and (2.21) for the analog correlation (repeated here for convenience)*

$E[z_1] = -j\,(2\,E_b/\,T_b)^{1/2}\exp\{j\,\theta_n\}(2/\pi)$, if $I_n = -1$ (2.20)*

and

$E[z_0] = j\,(2\,E_b/\,T_b)^{1/2}\exp\{j\,\theta_n\}(2/\pi)$, if $I_n = 1$. (2.21)*

Note that for $\theta_n = 0$ or π, the real parts of (2.20) and (2.21) are zero whereas the real parts of (4.6) and (4.7) are not. However, for large N_s we can approximate (4.6) and (4.7) using the binomial expansion by

$E[z_1] \approx -j\,(2\,E_b/\,T_b)^{1/2}\exp\{j\,\theta_n\}(2/\pi)$, if $I_n = -1$ (4.6a)

and

$E[z_0] \approx j(2\,E_b/\,T_b)^{1/2}\exp\{j\,\theta_n\}(2/\pi)$, if $I_n = 1$ (4.7a)

which are identical to the analog correlator outputs. The small discrepancy introduced into the correlator channels that do not contain the frequency of the current symbol is insignificant to the receiver performance for values of N of 64 or greater (recall that $N_s = N/4$ for MSK), however, it is important to use the exact formulas (4.6), (4.7) when debugging code.

5. Conclusions

The radix two fft may be used to generate and demodulate M-ary CPFSK with a wide range of rational modulation indices h=m/p so long as p is a power of two. For ordinary MSK, h = ½. The N>M point fft receiver is equivalent to a bank of M digital complex correlators that act on the receive baseband signal. The M complex receiver outputs may be used as the soft inputs to optimal decoding of CPFSK using a Viterbi decoder or they may be demodulated incoherently directly based on magnitude or coherently based on the amplitude of the real part as in a conventional receiver.

The primary advantage of this technique is its flexibility. With simple program control changes, the same DMT modem can be used for M-ary CPFSK, M-ary FSK-PSK, and OFDM as is illustrated by the MATLAB code in the Appendices.

6. References

[1] Moose, Paul H., "Generating and demodulating M-ary FSK-PSK using the FFT", **NPS Tech Rpt. EC-97-003** , March 1997.

[2] Proakis, John G., **Digital Communications,** Second Edition, McGraw-Hill, 1989.

[3] Minuto, Juan Carlos, "Simulation of adjacent channel interference in a UHF satellite system", MS Sys Engr Thesis, Naval Postgraduate School, Sept 1993.

[4]Lin and Costello, **Error Control Coding,** McGraw-Hill, 1983.

APPENDIX I

```
%                       DMT Modulator
%                       Written by: Paul H. Moose
%                       Naval Postgraduate School
%                       Monterey, CA
%
%       This m-file generates M-ary CPFSK, M-ary FSK or OFDM transmit waveforms.
%It is implemented using the inverse fft to digitally create the carrier
%frequencies required for each symbol.
%
%INPUTS:
%       type ='mfsk','cpfs' or 'ofdm' to specify the desired modulation
%       q1 = no. of bits carried by each of the M1-ary input characters
%       q2 = no. of bits carried by each of the M2-ary input characters
%       KK = no. of carriers used in the OFDM symbols. Set to 0 for
%               M-CPFSK     and M-FSK.
%       N = no. of points used in the ifft. Must be greater than KK for
%               OFDM. Should be at least 4*M (M=2^q) for M-CPFSK or M-FSK.
%       rb = symbol rate for M-FSK and CPFSK; number of points in guard interval
%               for OFDM.
%       del_f = frequency separation of carriers.
%
%       S = matrix of M-ary input characters in decimal integer
%         notation. The first row contains the M1-ary characters that will
%         be used for the FSK, CPFSK or OFDM. If there is a second row,
%       it contains the M2-ary characters for phase modulation of FSK-PSK.
%
%OUTPUTS:
%       x = complex baseband output sample sequence.
%       X = frequency domain array of modulation values. The columns are
%               length N and each column represents a transmission symbol.
%               x is formed from ifft(X).
%       MM = frequency domain array of modulation values used to form X.
%               For M-CPFSK or FSK, the columns of MM are of length M. For OFDM
%               the columns of MM are of length KK. In the case of M-CPFSK or FSK
%               only one of the rows of each column is non-zero, corresponding
%               to the frequency to be transmitted for that symbol. In
%               the case of OFDM, all of the rows of each column contain
%               modulation values to be transmitted. In the case of
%               M-FSK, q bits are transmitted with each symbol, while
%               in OFDM, Q*KK bits are transmitted with each symbol.
```

```
%       MP = the real output obtained from quadrature modulation of x
%               onto a carrier frequency fo (fo is currently set to
%               1200 in the program). (MP is automatically plotted if
%               there are fewer than 20 output symbols). For M-FSK only.
%
%SUBROUTINES REQD. :
%       freqa.m
%
% USEAGE:
%       [x,X,MM,MP]=dmtmod2(type,q1,q2,KK,N,rb,del_f,S)
%==========================================================
========

function [x,X,MM,MP]=dmtmod2(type,q1,q2,KK,N,rb,del_f,S)

%==========================================================
=========

%Initialize

[aa,cc]=size(S);

%==========================================================
=========
%==========================================================
=========

%M-FSK, CPFSK: Determine number of cycles of fundamental carrier and number of
%sample points to be used to account for the fractional cycle when the del_f
%is not an exact multiple of rb.


if type=='mfsk'
        %if rem(del_f,rb)==0 %Use this if del_f is a multiple of rb
        %       P=del_f/rb;
        %       Ns=N;
        %else % Use this when del_f is not a multiple of rb.
                cycles=del_f/rb;
                %P=fix(cycles)+1;
                P=ceil(cycles);
                fraction=P-cycles;
                Ns=N-round(fraction*N/P);
                Nss=N-fix(fraction*N/P);
        %end
```

```
%=====================================================================
===============
%
%Display the Sampling frequency , Exact DEL_F,),
% Carson's Rule BW, and Bitrate.
        rbb=N*del_f/(Nss*P);
        fs=Ns*rb
        DEL_F=fs*P/N
        %BW=rb+(2^q1-1)*DEL_F
        Ns
        Bitrate=q1*rb
        rbb
%=====================================================================
===============
%Generate array of ones properly spaced to give next integer number of cycles
% per symbol above the correct number of cycles.


        s=S(1,:);
        s=P*s+1;



%Now compute the phase modulation, if any using the second row of S

        if aa==2

                M2=2^q2;
                delphi=2*pi/M2;
                phi=delphi*(S(2,:));
                MP=exp(2*pi*j*phi);

        else
                MP=ones(1,cc);
        end

        for n=1:cc
                MM(s(n),n)=MP(n);
        end

%=====================================================================
========
% Locate carriers in the frequency domain array of N digital
```

```
% frequencies with carrier number one
%at frequency -M/2, carrier M/2 at frequency -1,carrier M/2+1
% at zero frequency and carrier M at frequency M/2-1.
% Minimum value for N is 2*M.

        X=freqa(N,MM);

%==============================================================
=========

%Take ifft of frequency domain array producing time domain array
%of cc symbols of N points each of which are one of M complex sinusoids.

        x=ifft(X);
        x=x(1:Ns,:);%Shorten the array to Ns points to remove the
                        %fractional cycle when del_f is not multiple
                        %of rb.
%==============================================================
========

elseif type == 'cpfs'

        h=del_f/rb
                %P=fix(cycles)+1;
        P=ceil(h)
        %fraction=P-cycles;
        Ns=round(h*N/(2*P));
        %Nss=N-fix(fraction*N/P);
        %end
%==============================================================
=================

%Display sampling freq, exact freq spacing, and exact mod index.
%
        fs=Ns*rb
        DEL_F=2*fs*P/N;
        Ns;
        h= 2*P*Ns/N;
        mm=P*Ns;
        pp=N/2;
        while rem(mm,2)==0 & pp>1
                mm=mm/2;
                pp=pp/2;
        end
        disp('    Ns            h           mm     pp    DEL_F    ')
        disp([Ns h mm pp DEL_F])
%==============================================================
```

```
===============
%Generate array of ones properly spaced to give next integer number of cycles
% per symbol above the correct number of cycles.


        s=S(1,:);
        I=2*s-(2^q1-1)



%Now compute the cumulative phase modulation.
        %cumsum(I)

                cumphi=[0 rem(pi*h*cumsum(I),2*pi)]

                loc =rem( P*I+N,N)+1

                X=zeros(N,cc);
        for n=1:cc
                X(loc(n),n)=exp(j*cumphi(n));
        end

%=============================================================
========
%Take ifft of frequency domain array producing time domain array
%of cc symbols of N points each of which are one of M complex sinusoids.

        x=ifft(X);
        x=x(1:Ns,:);%Shorten the array to Ns points to remove the
                    %fractional cycle when del_f is not multiple
                    %of rb.
%=============================================================
=============
%=============================================================
========
elseif type=='ofdm'
%=============================================================
==========
        fs=N*del_f
        BW=KK*del_f
        Bitrate=KK*q1*del_f
%=============================================================
```

```
==========
        %The serial symbol stream s is first inverse muxed into KK streams
        %that will be the rows of the matrix S. The columns of S
        %will be ofdm symbols.

        r=rem(cc,KK);
                if r~=0
                        disp('      ')
                        disp('Input being truncated by')
                        disp(r)
                        disp('symbols')
                end
        sof=S(1,1:cc-r);
        L=length(sof)/KK;
        Sof=reshape(sof,KK,L);
        [KK,L]=size(Sof);

%=====================================================================
=========

%Modulation values MM with amplitude one and one of 2^q=M equal
% phase values are differentially coded for each of the KK
% carriers. The first column (symbol) is one zero phase.
% The next columns are differentially coded in phase.


        dph=2*pi/2^q1;

        SD=cumsum(Sof)'; % Differentially code the phase values

        MM= exp(i*dph*SD);% Generate the modulation values.

        MM=[ones(KK,1) MM];% Add the reference modulation values.(Should
                            %change these from all ones in the future)

%=====================================================================
========

% Locate the modulation values in the frequency domain array
% of digital carriers

        X=freqa(N,MM);

%=====================================================================
```

```
==========
% Create the multiple ofdm carriers by executing the ifft and add the guard
% interval.

        x=ifft(X);

                if rb==0
                x=x;
                else
                x=[x(N-rb+1:N,:); x];
                end
end
%=====================================================================
============

% Quadrature modulate the baseband symbols onto an IF carrier frequency fo and
%if the modulation type is 'mfsk' or 'cpfsk'

%if type == 'mfsk'|'cpfs'
[rr,cc]=size(x);



fo=800
ko=fo*N/fs;
ko=floor(ko)  % The digital carrier frequency corresponding to fo
MP=x(:).';
nn=0:length(MP)-1;
MP=MP.*exp(2*pi*i*nn*ko/N);
MP=real(MP); %The real output signal

% Plot output automatically for short inputs
if cc<=20
        t=0:length(nn)-1;
        t=t/fs;
        plot(t,MP)
%end
end
```

APPENDIX II

```
%function   [Mr,Y,theta]=dmtdmd2(type,q,KK,N,rb,del_f,y)
%
%                   DMT De-modulator
%                   Written by: Paul H. Moose
%                   Naval Postgraduate School
%                   Monterey, CA.
%
%     This m-file demodulates M-ary CPFSK, FSK  or OFDM. It is implemented
% using with an fft to which is equivalent to a digital correlator
% for each of the carriers.
%
%INPUTS:
%     type = 'mfsk','cpfs' or 'ofdm' to specify the modulation
%     q =     no. of bits carrier in the M-ary CPFSK, FSK or no. of bits
%             carried by each of the OFDM carriers.
%     KK =    no. of carriers used in each OFDM symbol. Set to zero
%             for M-CPFSK or FSK.
%     N =     no. of points to be used in the fft.
%     rb =    symbol rate for M-CPFSK or FSK. Number of points in guard interval
%             for OFDM.
%     del_f = frequency separation of carriers
%     y =     matrix of input time domain symbols. Each column contains
%             complex baseband samples for one symbol.
%
%OUTPUTS:
%     Mr =    Matrix of recieved modulation values. Each column is a
%             vector containing the complex modulation values of the
%             M=2^q carriers for mfsk and cpfsk or the KK carriers for ofdm
%     Y =     Matrix of the fft of y after y has been zero padded to N
%             in the case of mfsk and cpfsk or after removing guard intervals
%             in the case of ofdm.
%     theta =         The phase states for cpfsk
%USEAGE:
%     function   [Mr,Y,theta]=dmtdmd2(type,q,KK,N,rb,del_f,y)
%
%=============================================================
========
function [Mr,Y,theta]=dmtdmd2(type,q,KK,N,rb,del_f,y)
%=============================================================
=========
%Initialize
```

```
[rr,cc]=size(y);
%=============================================================
==========
%=============================================================
==========
%M-FSK: Extend symbols to length N and take fft.

if type=='mfsk'
        if rem(del_f,rb)==0
                P=del_f/rb;
                Ns=N;
        else
                cycles=del_f/rb;
                P=fix(cycles)+1;
                fraction=P-cycles;
                Ns=N-round(fraction*N/P);
        end

        fs=Ns*rb

        DEL_F=fs*P/N

        ye=[y;zeros(N-rr,cc)];
        Y=fft(ye);


%=============================================================
========
% Extract KL digital carriers from the  frequency domain array of N digital
% carrier frequencies and place in the received array R.
% Of the KL freqs in R, 2^q are the matched filter outputs of the mfsk signal.
P
        KL=P*(2^q-1)+1;
        K=floor(KL/2)
        R=ifreqa(K,Y);

%=============================================================
========
%Sample the filter outputs to obtain the 2^q modulation values for each
%symbol.
        [rr,cc]=size(R);
        Mr=R(1:P:2^q*P,:);
```

```
%===============================================================
========
elseif type =='cpfs'

        h=del_f/rb;
        P=ceil(h);
        Ns=round(h*N/(2*P));
        fs=Ns*rb;
        DEL_F=2*fs*P/N;
        h=2*P*Ns/N;
        mm=P*Ns;
        pp=N/2;
        while rem(mm,2)==0 & pp>1
                mm=mm/2;
                pp=pp/2;
        end
        disp('   Ns            h     mm     pp      DEL_F  ')
        disp([Ns h mm pp DEL_F])

% Now compute the phase states

        if rem(mm,2) == 0

                theta = pi*h.*(0:pp-1);
        else
                theta = pi*h.*(0:(2*pp-1));
        end


        Y=fft(y,N);

        Mr=[Y(N-(2^q-1)*P+1:2*P:N-P+1,:); Y(P+1:2*P:(2^q-1)*P+1,:)];


%===============================================================
========
elseif type=='ofdm'

        %The precursor is removed from the input symbols and the fft is
        %computed

        y=y(rb+1:rb+N,:);

        Y=fft(y);
```

```
%============================================================
==============
% The modulation values of the KK digital carriers are extracted and
% placed in the columns of array R.

        K=floor(KK/2);

        R=ifreqa(K,Y);
        R=R(1:KK,:);

%============================================================
============
%Differentially decode the symbols in the time domain.
%       [rr,cc]=size(R);

                for l=1:cc-1
                        Mr(:,l)=R(:,l+1).*conj(R(:,l));
                end
%============================================================
===============



end
```

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center 8725 John J. Kingman Rd, STE 0944 Ft. Belvoir, VA 22060-6218	2
2.	Dudley Knox Library, Code 52 Naval Postgraduate School 411 Dyer Road Monterey, CA 93943-5101	2
3.	Research Office, Code 09 Naval Postgraduate School 589 Dyer Road Monterey, CA 93943-5138	1
4.	Chairman, Code EC Department of Electrical and Computer Engineering Naval Postgraduate School 833 Dyer Road Monterey, CA 93943-5121	1
5.	Dr. Paul H. Moose, Code EC/Me Department of Electrical and Computer Engineering Naval Postgraduate School 833 Dyer Road Monterey, CA 93943-5121	4
6.	Dr. Herschel H. Loomis, Jr., Code EC/Lm Department of Electrical and Computer Engineering Naval Postgraduate School 833 Dyer Road Monterey, CA 93943-5121	1
7.	Dr. Richard North, Code D855 NRad 53560 Hull Street San Diego, CA 92152	2